NLP homework 2: Semantic Role Labeling

Alessio Palma
Sapienza University of Rome
palma.1837493@studenti.uniroma1.it

1 Introduction

Semantic Role Labeling (SRL) is the task of assigning labels to arguments in a sentence that indicates their semantic role with respect to a predicate. A predicate defines an action or event, while an argument is a participant to the action or feature of the event and the semantic role is the relation of the arguments with respect to the predicate. SRL can be divided into 4 sub-tasks: predicate identification, predicate disambiguation, argument identification and argument classification; my work implemented the last two steps of this pipeline and the semantic roles are defined according to VerbAtlas (Di Fabio et al., 2019). I started from a simple BiLSTM model with GloVe word embeddings, then I progressively added one new component at a time on top of the model and only if it brought improvements on the English validation set then it was kept. The sections of this report follow the chronological order of the extensions that I added. The final model is a Transformer-based BiLSTM that also uses informations of POS tags embeddings, which has already been proved to be effective for this kind of task (Shi and Lin, 2019; Bae and Lee, 2022).

2 Data preprocessing

As any other machine learning task, I started from looking at the data. Since a sentence can have more than one predicate, I unrolled the samples in order to have at most one annotated predicate for each sample. So in the processed dataset I will have repeating sentences but with a different annotated predicate, for which the system will recognize the semantic role of the arguments. After this preprocessing, I have a total of 12641 train samples for the English dataset, 1130 train samples for the French one and 1085 train samples for the Spanish one.

3 English model architecture

This section describes the model developed for argument identification and classification on the English dataset. All the reported F1 scores are to be intended on the argument classification task on the English validation dataset, since it is the main evaluation metric for the homework.

3.1 Baseline

Given an input sentence as a sequence of words, each word needs to be transformed into a meaningful numerical representation, hence an embedding layer is used for this purpose, mapping each word into an higher-dimensional latent space. This layer is frozen and is initialized with the GloVe 300-dimensional word embeddings (Pennington et al., 2014), which better capture the semantic and syntactic relationships between words because they are trained on larger datasets. These embeddings are exposed in python through the Gensim library [1]. The vocabulary considered to build the embedding layer is the full set of words present in the dataset, this choice was made because there are not many word types (8857). After the embedding layer this model has a 3-layer bidirectional LSTM (BiLSTM) (Graves and Schmidhuber, 2005), which was the main component of my model for homework 1. BiLSTMs stack two LSTMs on top of each other: one will process the input sequence in forward order encoding the context before every token, the other will process it in reverse order encoding the context after each token. The two hidden representations obtained for each token are then concatenated in output, in this way the network knows both the context before and after each token. I also used Dropout (Hinton et al., 2012), a regularization technique that randomly zeroes neurons of the network during training, after each layer

[1] https://radimrehurek.com/gensim/

because the number of parameters is quite big and I wanted to avoid overfitting. At the end, a linear layer with Softmax activation function is used for classification. This simple model reaches an F1 score of **35.4%**. The performance is low because I am not feeding to the network the information on which is the predicate in the sentence, so it is not able to recognize arguments well. Hence, I added a predicate embedding layer and concatenated the embedding of each word with a 150-dimensional embedding of its predicate frame, reaching an F1 score of **84.5%**. I also tried fine-tuning the GloVe embeddings, but performance decreased so i left them frozen.

3.2 Part of speech tags

Part Of Speech (POS) Tagging is a main task in NLP that can often improve many downstream tasks. Adding POS tags to the input tokens is useful because of the intuition that some words are most likely to be semantic roles with respect to other ones (e.g adjectives and interjections alone are very rarely semantic roles, while nouns can very often be). Every POS was mapped into an embedding and concatenated to the respective word ∘ predicate embedding (∘ is the concatenation operator), the resulting vector is then fed as input to the BiLSTM. At this point I tried different dimensions for the POS embedding (see table 1) and the best one was 100. With this addition, the model reached an F1 score of **85.78%**.

3.3 Contextualized word embeddings

At this point, I replaced static word embeddings with contextualized word embeddings produced by a Transformer-based model. The contextualized embedding for a word is important because it will be different based on the context of the sentence, as oppose to non-contextualized embeddings (GloVe, Word2vec, ...) that will produce the same vector representation regardless of the context. In this way, different senses of a word are not collapsed into the main sense, but are all captured based on the surrounding words in the sentence. Since the advent of the Transformer (Vaswani et al., 2017), large pre-trained language models have become the de facto standard for obtaining contextualized word embeddings. I tested different Transformer models (see table 2) exposed through Hugging Face [2] and the best performing was the non fine-tuned RoBERTa (Liu et al., 2019), which is an encoder Transformer based on the same architecture as BERT (Devlin et al., 2018), but trained on a lot more data (10x) and using a BPE tokenizer instead of a WordPiece tokenizer, with a larger subword vocabulary (50k vs 32k). To obtain an embedding for each word, I averaged the last 4 layers of the Transformer model and averaged the sub-tokens belonging to the same words, using the Transformers Embedder library [3]. Since RoBERTa is case-sensitive and until now I was working with lowercased dataset, I removed the lowercasing and reached an F1 score of **87.86%**.

3.4 Lemmas

The lemma is the dictionary form of a word, I tried concatenating a 300-dimensional lemma embedding to the current token representation (contextualized word embedding ∘ predicate embedding ∘ POS embedding), believing that grounding each token also to a static latent meaning could improve the performance, but it was not the case (see table 3), so I removed the lemma embedding to have less parameters. The fact that performance remained almost the same demonstrates that contextualized word embeddings already include all the semantic information that a static embedding could give.

4 Training

After all the additions I performed one last coarse-grained grid search over hyperparameters as reported in table 4, the final model reaches an F1 score of **88.81%** and you can see its confusion matrix in image 2. It is trained using Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001 and no L2 regularization. In order to reduce overfitting, every experimented model is trained for 60 epochs using early stopping with 9 epochs of patience, actually causing a model to never be trained for more than 40 epochs. Moreover, the saved weights are the one that produced the best validation F1 score during the training, this avoids saving noisy weights and further reduces overfitting, as you can see in image 1. Batch size was fixed to 32 and all the models were trained on a local GPU. The loss to be minimized is the Cross Entropy.

[2] https://huggingface.co/docs/transformers/index

[3] https://github.com/Riccorl/transformers-embedder

5 Other languages

In order to perform the SRL task also on the given French and Spanish datasets, I decided to compare various approaches based on the same architecture of the final English model, which are:

1. Substitute the Transformer module with a language specific one and retrain the whole model;
2. Substitute the Transformer module with a language specific one, transfer weights from the English model for all the remaining layers and fine-tune;
3. Substitute the Transformer module with a language specific one, transfer weights from the English model for all the remaining layers except the last linear layer and fine-tune.

The best approach proved to be the second one for both French and Spanish, demonstrating the fact that pre-training on English can effectively improve the performance of the model on other languages, by transferring some semantic and syntactic knowledge that is in common between various languages. Training procedure is the same described in section 4.

5.1 French

For French the choice of the language model was very simple because CamemBERT (Martin et al., 2020), a RoBERTa trained on a large French corpus, stands alone. The results on this dataset are reported in table 5, the best model achieves an F1 score of **77.04%** and its confusion matrix can be seen in image 3.

5.2 Spanish

For Spanish the choice of the language model was harder because it wasn't clear if there is one Transformer that is explicitly superior to the others. After some tests, the Spanish RoBERTa developed in the MarIA project (Fandiño et al., 2022) proved to be the best. The results can be seen in table 6, the best model achieves an F1 score of **78.68%** and its confusion matrix is reported in image 4.

References

Jangseong Bae and Changki Lee. 2022. Korean semantic role labeling with bidirectional encoder representations from transformers and simple semantic information. *Applied Sciences*, 12(12).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637, Hong Kong, China. Association for Computational Linguistics.

Asier Gutiérrez Fandiño, Jordi Armengol Estapé, Marc Pàmies, Joan Llop Palao, Joaquin Silveira Ocampo, Casimiro Pio Carrino, Carme Armentano Oller, Carlos Rodriguez Penagos, Aitor Gonzalez Agirre, and Marta Villegas. 2022. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68.

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4.

Geoffrey Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint*, arXiv.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

POS embedding size	F1 score
25	85.57%
50	85.45%
100	**85.78%**
150	85.58%
300	85.40%

Table 1: Different POS embedding dimensions with relative results.

Model	Fine-tuning LR	F1 score
BERT-base-uncased	•	86.97%
BERT-base-uncased	1e-5	85.16%
BERT-base-uncased	3e-5	84.25%
RoBERTa-base	•	**87.74%**
RoBERTa-base	1e-5	86.28%
RoBERTa-base	3e-5	85.45%

Table 2: Comparison of F1 scores obtained using different pre-trained language models, dataset was lowercased at this stage. Where • is present, it means it was not fine-tuned.

Initialization	Fine-tuning LR	F1 score
Random	1e-3	**87.86%**
GloVe	•	87.77%
GloVe	1e-5	87.68%

Table 3: Comparison of F1 scores obtained adding different lemma embeddings. Where • is present, it means the layer was frozen.

Hyperparameter	Values
BiLSTM's hidden dim	**256** \ 512
POS embedding size	**50** \ 100
Predicates embedding size	150 \ **300**
Learning rate	**1e-3** \ 5e-4
L2 regularization	**0** \ 1e-5

Table 4: Final grid search was performed on these hyperparameters. Final model's hyperparameters are highlighted in bold.

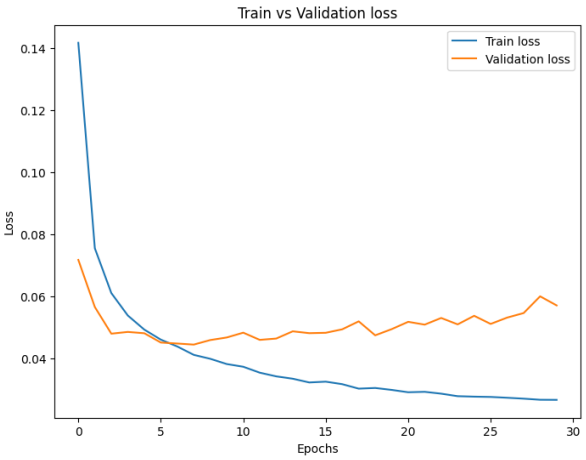

Figure 1: Train vs validation loss of the final English model. The best performing weights were saved at epoch 26, patience was consumed at epoch 30.

Transformer	Transfer learning	LR	F1 score
CamemBERT-base	None	1e-3	74.73%
CamemBERT-base	All the remaining layers	1e-3	**77.04%**
CamemBERT-base	All the remaining layers	1e-4	76.35%
CamemBERT-base	All the remaining layers	1e-5	74.97%
CamemBERT-base	All the remaining layers except classifier	1e-3	75.56%
CamemBERT-base	All the remaining layers except classifier	1e-4	76.61%
CamemBERT-base	All the remaining layers except classifier	1e-5	64.85%

Table 5: Analysis of the French model.

Transformer	Transfer learning	LR	F1 score
BETO-base-cased	None	1e-3	71.89%
SpanBERTa-base-cased	None	1e-3	71.88%
BERTIN-base	None	1e-3	71.44%
MarIA-RoBERTa-base	None	1e-3	75.21%
MarIA-RoBERTa-base	All the remaining layers	1e-3	**78.68%**
MarIA-RoBERTa-base	All the remaining layers	1e-4	76.87%
MarIA-RoBERTa-base	All the remaining layers	1e-5	76.29%
MarIA-RoBERTa-base	All the remaining layers except classifier	1e-3	73.83%
MarIA-RoBERTa-base	All the remaining layers except classifier	1e-4	76.46%
MarIA-RoBERTa-base	All the remaining layers except classifier	1e-5	62.94%

Table 6: Analysis of the Spanish model.

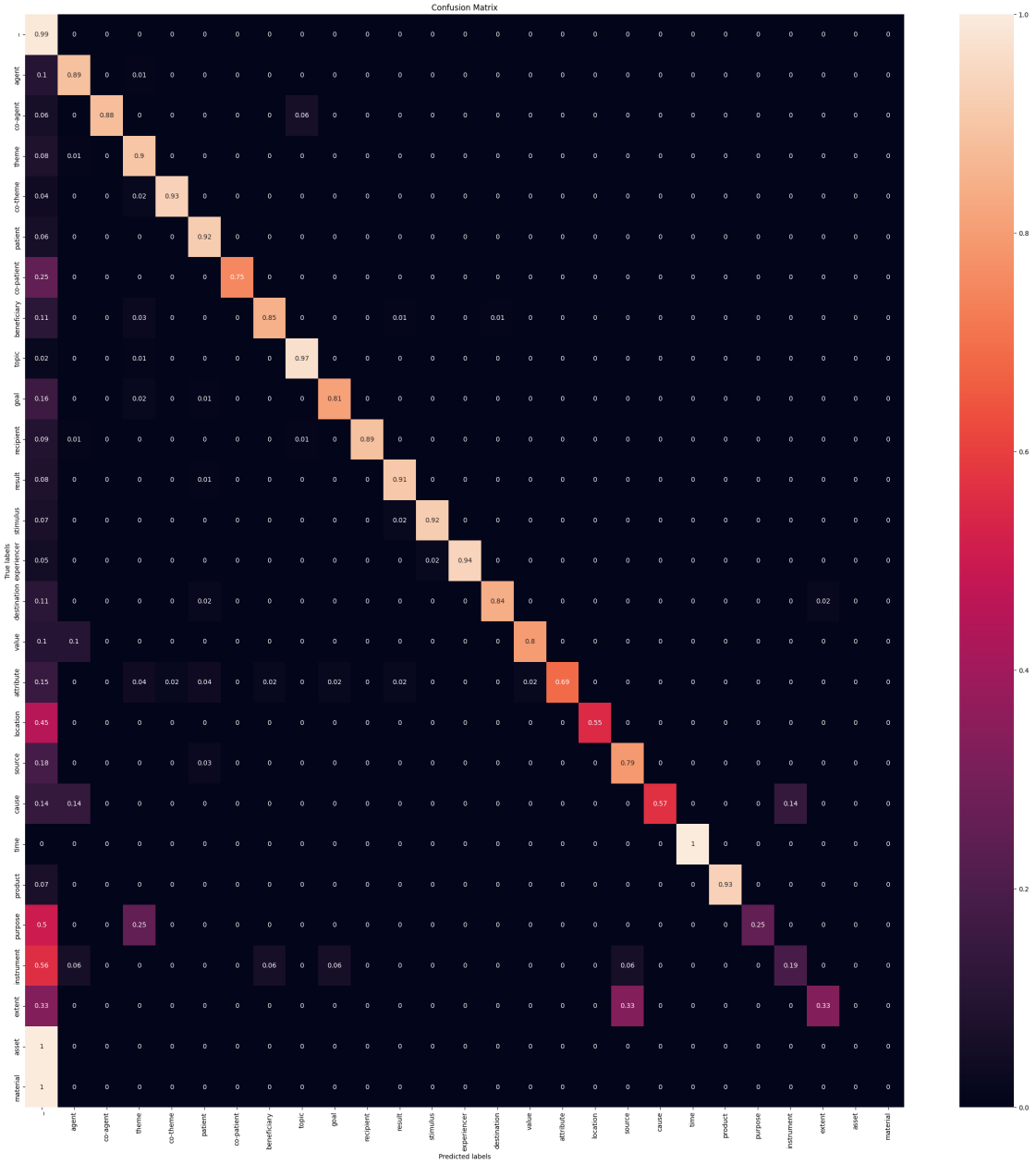

Figure 2: Confusion matrix of the final English model, we can see that less represented roles like instrument, purpose, asset and material are rarely recognized (the last two compare only 1 time in the validation set). Most common roles like theme and agent are very well recognized.

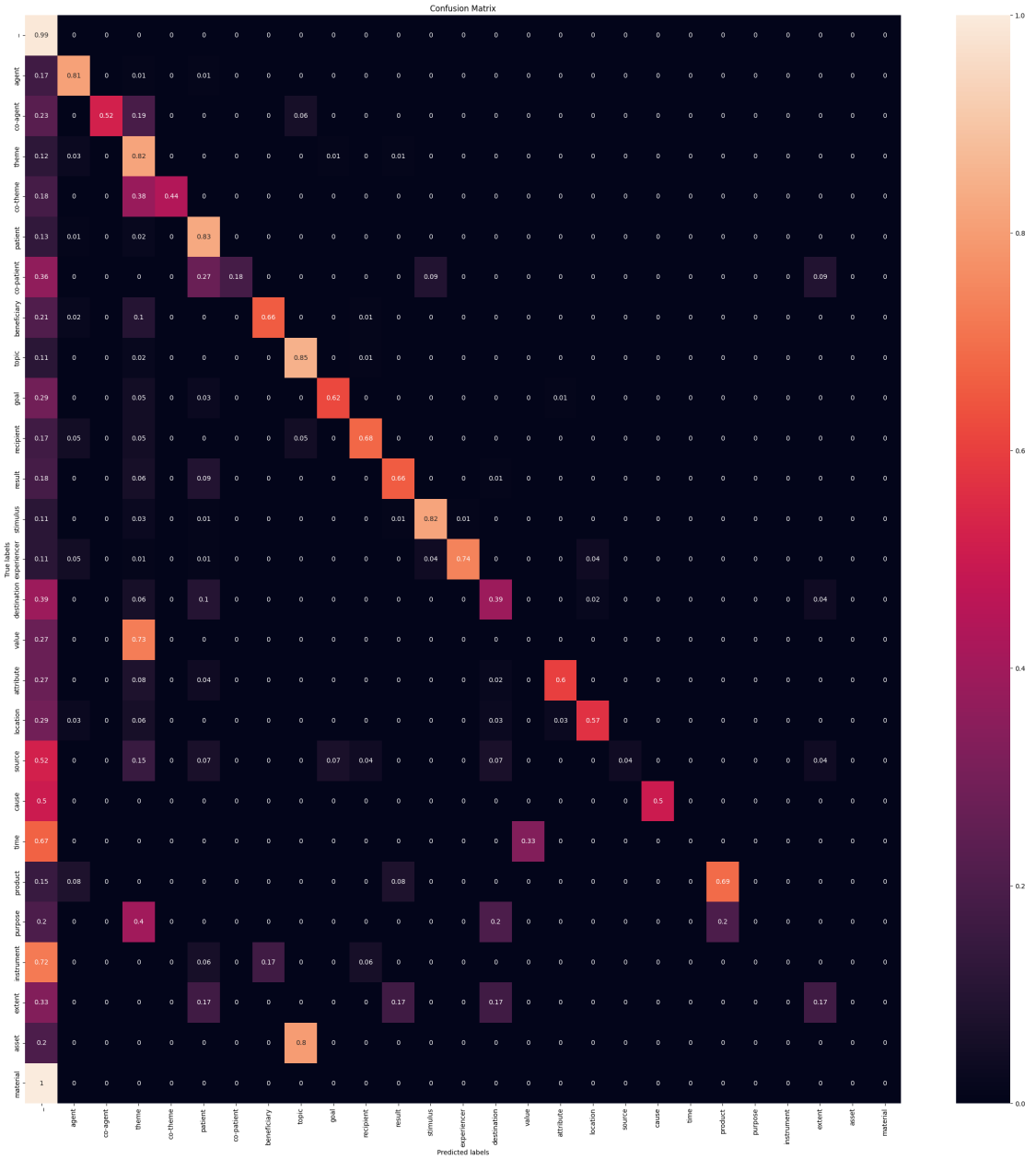

Figure 3: Confusion matrix of the final French model.

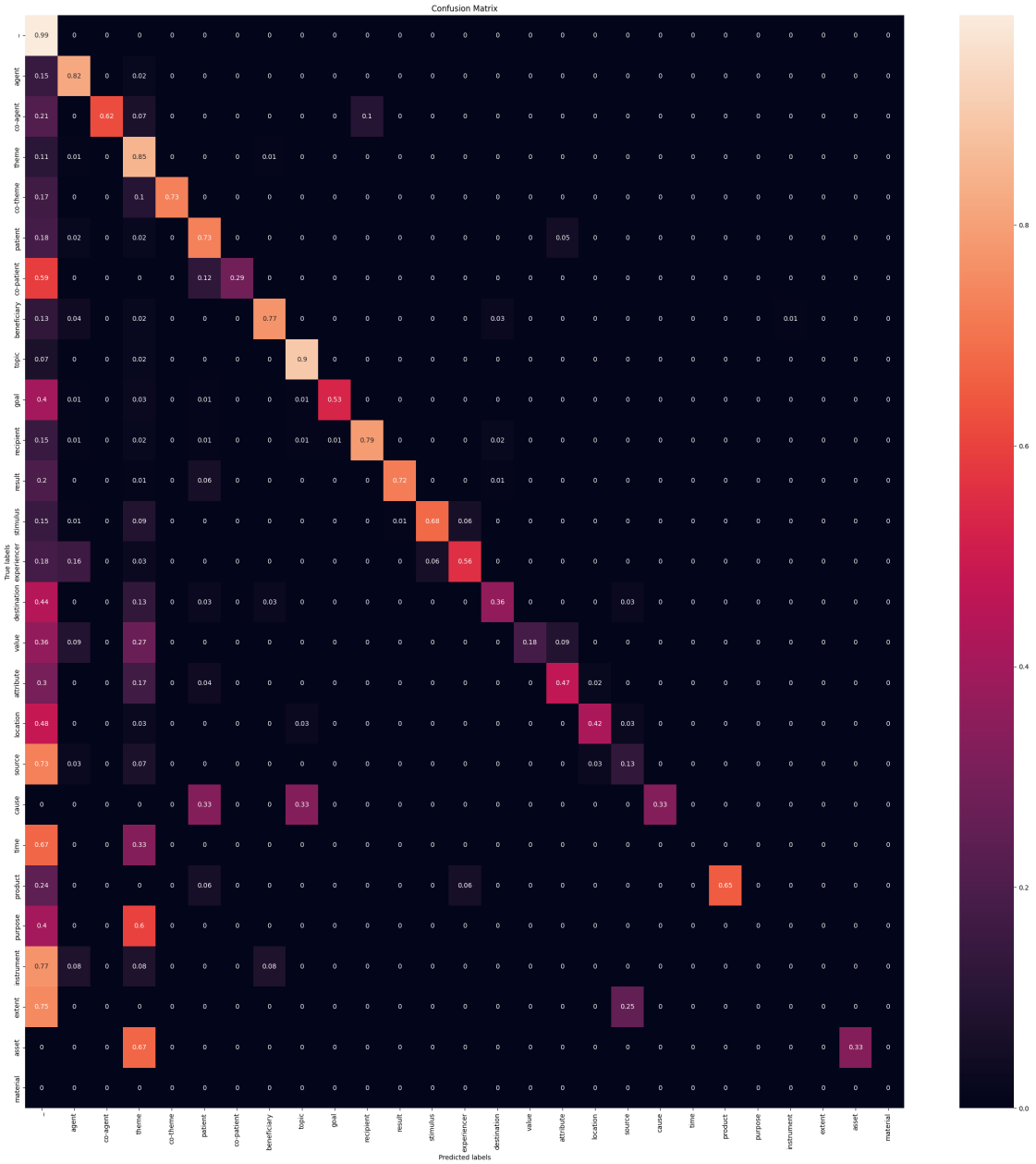

Figure 4: Confusion matrix of the final Spanish model.